DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

TP1

Algoritmos y Estructuras de Datos III

Integrante	LU	Correo electrónico
Sebastián Fernandez Ledesma		
Fernando Gasperi Jabalera	56/09	`fgasperijabalera@gmail.com`
Maximiliano Wortman		
Santiago Camacho	110/09	`santicamacho90@gmail.com`

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
http://www.fcen.uba.ar

Índice

1. Problema 1: Puentes sobre lava caliente

1.1. Presentación del problema

1.2. Resolución

1.2.1. Algoritmo

1.2.2. Pseudocódigo

1.3. Demostración

Vamos a definir un salto como un entero natural mayor que 0 y menor que la cantidad de tablones que el participante actual puede saltar de a una sola vez. Nuestra implementacion recorre el puente dando saltos, garantizando que en cada salto, la distancia recorrida es máxima. Es decir, no existe otro salto tal que la distancia desde donde estamos parados es mayor a la del salto actual y el tablón en el que caes no esta roto.
$Distancia$ es un Nat > 0.
$Salto$ es Nat tal que $\forall s : Salto, s > 0 \wedge s \leq Distancia$

Vamos a tratar de probar que dado una secuencia de saltos, si para cada salto s, s es un "salto maximoz si la sumatoria de saltos es mayor a la cantidad de tablones del puente, entonces nuestra secuencia es solucion del problema.

Dada un Sec<Salto> se.

$(\forall i : Nat, i < se.long)(esMax(se_i) \wedge \sum_{j=0}^{se.long-1} se_j = puente.long) \implies$
$\nexists(se' : Sec < Salto >)/se.long < se'.long \wedge \sum_{j=0}^{se'.long-1} se'_j \geq puente.long)$ //

Llamemos a sMax a la secuencia de saltos maximos obtenida. Supongamos que existe secuencia s de saltos tal que la cantidad de elementos de s es menor a sMax y la sumatoria de saltos es igual o mayor a la cantidad de tablones. Bueno en particular, existe al menos un salto s_i, tal que s_i, es mayor a $sMax_i$, ya que si todos los s_i, son menores a su correspondiente $sMax_i$, entonces la sumatoria de sMax es mayor que la sumatoria de s. (comprobar esto ad-hoc, probablemente sale por induccion). Bueno, supongamos que agarro el primero de todos los s_i, que es mas grande que su correspondiente $sMax_i$. Hasta ese momento las dos subsecuencias (desde el principio hasta el elemento i) pesan lo mismo, entonces s_i esta parado en el mismo lugar y hace un salto mas grande que el salto maximo ($sMax_i$), lo cual es absurdo. Por lo tanto queda comprobado que ese s_i, no puede existir y la solucion es máxima.

LIMPIADA DE CARA FER

Definimos un salto s como un natural mayor a 0 y menor o igual a la distancia máxima que puede recorrer el participante, de sólo un salto, medida en tablones

$$s \in Saltos \Leftrightarrow (s \in \mathbb{N}_{>0} \wedge s \leq dist_{max})$$

Definimos un puente como una función $p : \mathbb{N}_{>0} \to \mathbb{N}$

$$p(i) = \begin{cases} 1 & \text{i} \leq 0 \\ 0 & \text{i} > \#tablones \\ 0 & \text{i} > 0 \wedge i \leq \#tablones \wedge i \in Tablones \\ 1 & \text{i} > 0 \wedge i \leq \#tablones \wedge i \notin Tablones \end{cases}$$

Para cada posición i del puente definimos su salto máximo s_{max} como

$$s_{max} = max\{n \in \mathbb{N}_{>0} \mid n \leq dist_{max} \wedge \neg p(n)\}$$

Nuestra implementacion recorre el puente dando saltos, garantizando que en cada salto, la distancia recorrida es máxima. Es decir, no existe otro salto tal que la distancia desde donde estamos parados es mayor a la del salto actual y el tablón en el que caes no esta roto.
Distancia es un Nat > 0.
Salto es Nat tal que $\forall s : Salto, s > 0 \wedge s \leq Distancia$

Vamos a tratar de probar que dado una secuencia de saltos, si para cada salto s, s es un "salto maximoz si la sumatoria de saltos es mayor a la cantidad de tablones del puente, entonces nuestra secuencia es solucion del problema.

Dada un Sec<Salto> *se*.

$(\forall i : Nat, i < se.long)(esMax(se_i) \wedge \sum_{j=0}^{se.long-1} se_j = puente.long) \implies$
$\nexists(se' : Sec < Salto >)/se.long < se'.long \wedge \sum_{j=0}^{se'.long-1} se'_j \geq puente.long)$ //

Llamemos a sMax a la secuencia de saltos maximos obtenida. Supongamos que existe secuencia s de saltos tal que la cantidad de elementos de s es menor a sMax y la sumatoria de saltos es igual o mayor a la cantidad de tablones. Bueno en particular, existe al menos un salto s_i, tal que s_i, es mayor a $sMax_i$, ya que si todos los s_i, son menores a su correspondiente $sMax_i$, entonces la sumatoria de sMax es mayor que la sumatoria de s. (comprobar esto ad-hoc, probablemente sale por induccion). Bueno, supongamos que agarro el primero de todos los s_i, que es mas grande que su correspondiente $sMax_i$. Hasta ese momento las dos subsecuencias (desde el principio hasta el elemento i) pesan lo mismo, entonces s_i esta parado en el mismo lugar y hace un salto mas grande que el salto maximo ($sMax_i$), lo cual es absurdo. Por lo tanto queda comprobado que ese s_i, no puede existir y la solucion es máxima.

1.4. Análisis de complejidad

1.5. Test de complejidad

1.6. Testing

2. Problema 2: Horizontes lejanos

2.1. Presentación del problema

2.2. Resolución

2.2.1. Algoritmo

2.2.2. Pseudocódigo

```
input: edificios
while quedan edificios do
  if empieza edificio then
    registro el edificio como abierto
    if altura del edificio es mayor a la del contorno then
      agrego la altura del edificio al contorno
    end if
  else
    saco al edificio de los abiertos
    if este edificio le daba la altura al contorno then
      agrego la altura del edificio abierto que le siga en altura al contorno
    end if
  end if
end while
return contorno
```

2.3. Demostración

2.4. Análisis de complejidad

2.5. Test de complejidad

2.6. Testing

3. Problema 3: Biohazard

3.1. Presentación del problema

3.2. Resolución

3.2.1. Algoritmo

3.2.2. Pseudocódigo

3.3. Demostración

3.4. Análisis de complejidad

3.5. Test de complejidad

3.6. Testing